(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 September 2001 (27.09.2001)

PCT

(10) International Publication Number
WO 01/71960 A1

(51) International Patent Classification[7]: H04K 1/00

(21) International Application Number: PCT/US01/08315

(22) International Filing Date: 16 March 2001 (16.03.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

60/190,481	18 March 2000 (18.03.2000)	US
09/563,664	2 May 2000 (02.05.2000)	US
60/257,822	21 December 2000 (21.12.2000)	US

(71) Applicant *(for all designated States except US)*: **DIGIMARC CORPORATION** [US/US]; 19801 SW 72nd Avenue, Suite 250, Tualatin, OR 97062 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **LEVY, Kenneth, L.** [US/US]; 110 NE Cedar Street, Stevenson, WA 98648 (US). **DECKER, Stephen, K.** [US/US]; 2530 Orchard Hill Place, Lake Oswego, OR 97035 (US).

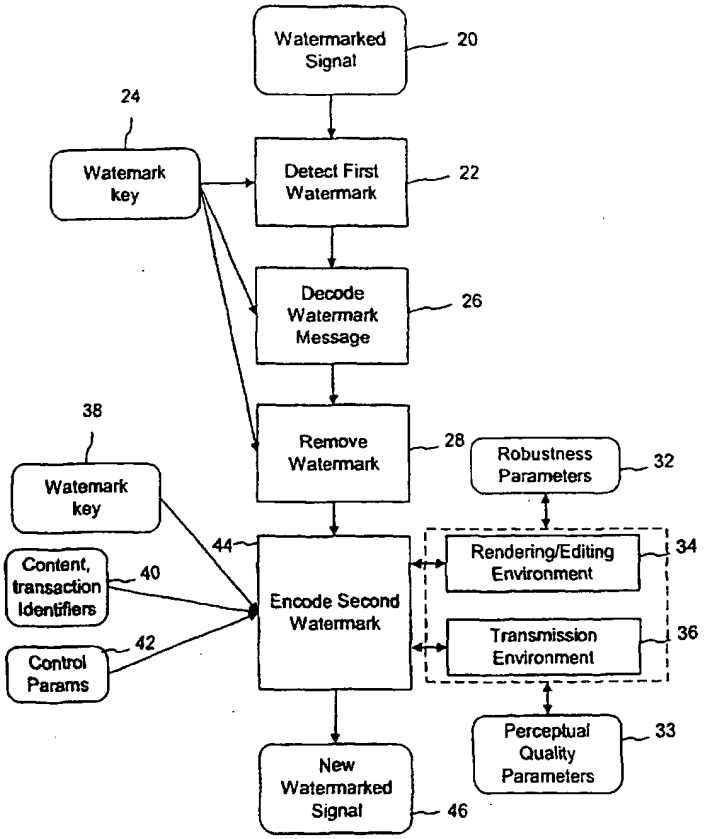
(74) Agent: **MEYER, Joel, R.**; Digimarc Corporation, 19801 SW 72nd Avenue, Suite 250, Tualatin, OR 97062 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian

[Continued on next page]

(54) Title: TRANSMARKING, WATERMARK EMBEDDING FUNCTIONS AS RENDERING COMMANDS, AND FEATURE-BASED WATERMARKING OF MULTIMEDIA SIGNALS



(57) Abstract: A watermarked media signal (20) is transmarked to adapt the watermark to the robustness and perceptibility constraints of the new environment (34). A first watermark is detected (22) in the media signal and the message information from the first watermark is embedded into a second watermark (44), before the media signal undergoes a transformation process. The second watermark (44) is adapted to survive the transformation process. In addition, a watermark embedding command is included in a set of rendering commands used during the process of creating a media object to specify how the media object is to be rendered. The watermark embedding command includes an identifier used to link to customer or related content information, the customer's web site, the intensity at which to embed the watermark, areas not to embed, batch processing options, printing preferences for images, watermark embedding methods to use on different media types, formats or different parts of the media object, and desired rendering quality.

WO 01/71960 A1

patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

Published:

— *with international search report*

Transmarking, Watermark Embedding Functions as Rendering Commands, and Feature-Based Watermarking of Multimedia Signals

Related Application Data

This patent application claims the benefit of U.S. Provisional Patent Application No. 60/190,481, entitled Embedded Data and Data Scrambling Improvements, filed March 18, 2000 by Ken Levy, which is incorporated by reference. This patent application also claims the benefit of U.S. Provisional Patent Application No. 60/257,822, entitled Watermark Systems and Methods, filed December 21, 2000, by Ken Levy et al. which are hereby incorporated by reference.

This patent application is also a continuation in part of U.S. Patent Application No. 09/563,664, entitled Connected Audio and Other Media Objects, filed May 2, 2000, by Ken Levy and Geoff Rhoads, which is hereby incorporated by reference.

This patent application is related to U.S. Patent Application 09/629,401, entitled Management of Document and Other Objects Using Optical devices, filed August 1, 2000, by Seder, Carr, Perry, Graham, and Rhoads, which is hereby incorporated by reference.

This patent application is also related to U.S. Patent Application 09/706,505, entitled Batch Identifier Registration and Embedding in Media Signals, filed November 2, 2000, by McKinley and Hein, which is hereby incorporated by reference.

This patent application is related to U.S. Patent Application No 09/404,292, by Ken Levy and assigned to AIPL, filed on September 23, 1999, which claims the benefit of U.S. Provisional Application Nos. 60/101,851 and 60/110,683 filed on 9/25/98 and 12/2/98, respectively, by Ken Levy, which are hereby incorporated by reference.

Technical Field

The invention relates to multimedia signal processing, and specifically, steganography, digital watermarking and data hiding.

Background and Summary

Digital watermarking is a process for modifying physical or electronic media to embed a machine-readable code into the media. The media may be modified such that the embedded code is imperceptible or nearly imperceptible to the user, yet may be detected through an automated detection process. Most commonly, digital watermarking is applied to media signals such as images, audio signals, and video signals. However, it may also be applied to other types of media objects, including documents (e.g., through line, word or character shifting), software, multi-dimensional graphics models, and surface textures of objects.

Digital watermarking systems typically have two primary components: an encoder that embeds the watermark in a host media signal, and a decoder that detects and reads the embedded watermark from a signal suspected of containing a watermark (a suspect signal). The encoder embeds a watermark by altering the host media signal. The reading component analyzes a suspect signal to detect whether a watermark is present. In applications where the watermark encodes information, the reader extracts this information from the detected watermark.

Several particular watermarking techniques have been developed, and, for robust watermarks, the goal is to design an imperceptible watermark that survives transformation. However, this cannot always be accomplished. The reader is presumed to be familiar with the literature in this field. Particular techniques for embedding and detecting imperceptible watermarks in media signals are detailed in the assignee's co-pending application serial number 09/503,881 and US Patent 5,862,260, which are hereby incorporated by reference. Watermarking techniques particularly adapted to graphic art and halftone images are set forth in U.S. Patent Application Nos. 09/074,034, entitled Methods and Systems for Watermark Processing of Line Art Images, 09/689,226, entitled Halftone Watermarking and Related Applications, and 60/263,987, entitled Halftone Primitive Watermarking and Related Applications, which are hereby incorporated by reference.

In watermarking applications and related literature, digital watermarks are classified as robust, fragile and semi-fragile. A robust watermark refers to a watermark that is designed to survive typical and even malicious processing of the watermarked

signal that distorts the watermarked signal and makes it more difficult to reliably detect and read the watermark. A fragile watermark refers to a watermark where the watermark degrades in response to certain forms of processing like printing copying, scanning, compression, etc. Fragile watermarks are typically used in authentication application to detect tampering of a signal. Semi-fragile watermarks combine the concepts of fragile and robust watermarks. These types of watermarks are designed to survive certain types of processing like compression, yet detect tampering like cropping or swapping of signals. Fragile and semi-fragile watermarks may be used to trigger certain actions or control usage of the watermarked content when degradation of the fragile watermark is detected.

In digitally watermarking media signals, such as audio, still images and video, there are a number of challenges and trade-offs. One challenge is to embed the watermark so that it is sufficiently robust for the particular set of attacks anticipated for the application, while making sure that the watermark is sufficiently imperceptible for that application. For some applications, it is not possible to fully anticipate the types of processing that a media object will encounter, even before it is distributed. For example, a music track may be produced and distributed in a number of different formats (different compression rates, different compression codecs, different broadcast formats, etc.). Each of these formats may degrade or distort the watermark differently. In addition, the music track may be rendered using high fidelity audio equipment, or lower quality equipment, giving rise to different perceptual quality constraints. In particular, lower quality rendering enables the watermark to be embedded more robustly because perceptibility constraints on the watermark are less stringent. The same is true for video signals, like movies, television programming, advertisements, etc.

In the case of still images, an image may undergo transformations, such as compression, color conversion, halftoning, etc. before it is finally printed or rendered. Consider, for example, graphic art used in advertisements, packaging, and brochures. Such art imagery may include a collection of a raster images that are combined to form a final image. For a particular design project, the graphic artist creates a piece of graphic art for a customer, typically including a collection of constituent images in

different formats. Some of the images may be line art, vector graphics, color halftone or color multi-level per pixel images (in color formats like RGB, CMYK or YUV). The entire image product is described in a job ticket that encapsulates the rendering functions to control the assembly of the constituent images and the printing process.

The customer may want to apply a watermark to the final image product for a variety of applications, such as inserting a customer identifier for tracking purposes, linking the image to the customer's web site, etc. There are two main problems, potentially inter-related. One problem occurs with the content flow and timing of adding the watermark flow. Another problem occurs with adding watermarks to vector graphics. The stage at which the watermark message payload and embedding parameters are defined may not always be the appropriate stage to embed the watermark in the host signal. One place to embed the message payload of the watermark into the graphic art is in the raster interface processing (RIP) stage. In this stage, the constituent images are assembled and converted to a particular halftone image format compatible with the printer. The halftone image format includes one or more color planes of pixel elements that specify the presence or absence of ink at corresponding pixel locations. The RIP stage usually occurs at the Pre-Press house or Printer, and requires the person with the most critical eye for color. In addition, this stage, by definition, results in a complete raster image. The watermark can be defined for vector graphics (or line-art), but is ultimately embedded in a raster image when printed with common modern equipment. The customer doesn't usually interact with the Pre-Press house or Printer, except to possibly proof the image. In addition, these locations are under terrible time and cost constraints and do not want to deal with inefficient and costly customer interactions. Finally, many graphic art pieces contain little or no raster sections; thus, the watermark cannot be added before the art is rasterized at the RIP stage. Despite the difficulty of watermarking prior to rasterizing for printing, it is often necessary to preview the watermarked final image product on a display screen, or desktop printer, which poses the problem of how to embed the watermark for previewing.

If the graphic artist has to add the watermark before the Pre-Press house or Printer, the graphic artist must rasterize the image. This causes two problems. First,

the graphic artist must now deliver a file consisting of a large number of bits (i.e. size). Second, the graphic artist is not the best person to deal with the color management required to produce a quality image.

The difficulty is that the customer is already working with the graphic artist and wishes to define the contents of the watermark, but the watermark is ultimately embedded in the rasterized image in the Pre-Press house or Printer. A similar problem exists for other media types like audio and video, where the watermark payload is specified at a stage different than the most appropriate stage for embedding the watermark in the content.

If the image file is a vector graphic, whether rendered for printing as described above, or distributed electronically such as on the web, a participant such as the owner, may want to watermark the vector graphic. The participant wants that watermark to be embedded in the rendered image whenever the vector file is rendered, such as on a computer screen, possible within a wed browser or printer. This allows illegitimate copies, such as copies made with a print screen function, to be identified.

A method for controlling watermark embedding in a media object through the use of a watermark embedding command is described below. In the process of creating the media object, the method includes a watermark embedding command among a set of one or more rendering commands that specify how the media object is to be rendered. For example, certain media signal formats like PCL, PDF, or postscript for images, MIDI and structured audio for audio signals, and MPEG-4 and MPEG-7 for audio and video signals, include descriptors that control how a particular media signal is to be rendered. The watermark embedding command includes a combination of the following items: an identifier used to link to customer or related content information, the customer's web site or store, the intensity at which to embed the watermark, areas not to embed, batch processing options, printing preferences for images, watermarking embedding methods to use on different media types, formats, or different parts of the media object, and desired rendering quality.

The watermark embedding command enables the customer or creator to specify watermark message payload and embedding parameters and preferences, and enables the rendering device to embed the watermark appropriately for a particular rendering

process. In the case of graphic art, the customer can preview the watermarked content on the graphic artist's monitor or inexpensive printer, which rasterizes the image for display, embeds the watermark in response to the command, and renders the watermarked image. In addition, the Pre-Press house or Printer, can add and modify the watermark without interacting with the customer, thereby saving time and money.

In general, the watermark embedding command includes the message payload to be embedded and rules or links to how to embed these bits. Thus, the watermark function is implemented according to the desired embedding method when the graphic art is rendered, such as on the screen, printed proofs or final printing plates.

This method is extended to other types of media objects, including audio or music tracks, video sequences, etc.

The advantages of watermark embedding commands include the following: watermarks can be embedded in rendering description content, such as vector graphics, MIDI, and structured MPEG audio and video. In addition, watermarks can be embedded at a time and location separate from where and when the watermark and content is rendered. This reduces costs by allowing proper interaction between the content owner and creators, who have different responsibilities and skills.

The invention further provides methods and related systems, devices and software for transmarking media signals. Transmarking relates to converting auxiliary data embedded in a media signal from one digital watermark format to another. It is used in processes that transform the media signal, such as compression, broadcast, editing, rendering, etc., to change the characteristics of the embedded watermark so that the watermark has improved robustness or perceptibility characteristics for its new environment. In some cases, transmarking can be extended to cases where out-of-band data file the header or footer of a media file, or other metadata provided with the media file is transmarked into a watermark or is derived from a watermark. Thus, the watermarks appear to be robust to all transformations.

One aspect of the invention is a method of transmarking a media signal previously embedded with a first digital watermark using a first digital watermark embedding method. This transmarking method detects the first digital watermark in the

media signal. It then embeds message information from the first digital watermark into a second digital watermark in the media signal before the media signal undergoes a transformation process. The second digital watermark is adapted to survive the transformation process.

Another aspect of the invention is another method of transmarking a media signal. This method detects the first digital watermark in the media signal, converts the media signal to a different format, and embeds message information from the first digital watermark into a second digital watermark in the converted media signal. The second digital watermark is adapted to robustness or perceptibility parameters associated with the new format.

Further features will become apparent with reference to the following detailed description and accompanying drawings.

Brief Description of the Drawings

Figure 1a: Diagram for embedding a feature-based watermark to ease searching a large image for a small watermarked area.

Figure 1a: Diagram for retrieving a feature-based watermark to ease searching a large image for a small watermarked area.

Figure 2: This figures shows a pseudo-random noise array that can be used to determine the scaling and rotation of an image via autocorrelation.

Figure 3: This figure demonstrates the art of slowing the transition between embedding auxiliary 0's and 1's.

Figure 4a: This figure shows the grid used to embed an autocorrelation-based watermark.

Figure 4b: This figure demonstrates how to skip embedding data in some blocks to find the orientation of an autocorrelation-based watermarked image. X's represent watermarked blocks; thus, blocks without Xs are not watermarked.

Figure. 5 is a diagram illustrating a transmarking process where a first digital watermark in a media signal is transmarked into a second digital watermark in the media signal.

Fig. 6 is a diagram illustrating a watermark embedding function and rendering description file.

Fig. 7 is a diagram illustrating a process for embedding watermarks in media objects using watermark embedding commands.

Detailed Description

Feature Based Watermark

When using noise reduction techniques, such as Weiner filtering or spectral subtraction, you can obtain the embedded watermark as noise. This noise represents the sum of all the watermark layers. This noise can be re-scaled and embedded in other images such that they impersonate the original image.

However, when embedding another noise layer that consists of local arrays around the largest N (maybe 5) features, such as peak of the derivative, of the image, this attack can be stopped. The idea is similar to US Patents Serial Numbers 5,809,160 and 5,930,377 invented by Powell and Nitzberg and assigned to Digimarc, included herein by reference. When using peaks, they should have a certain average slope around them. Alternatively, you could use the peak of the derivative of the image since an edge correlates to this peak and edges are good places to hide data.

To this end, when all the noise layers are moved from one image to the other as one grouped noise, as done with this copy attack, the new features most likely will not align with the old features. As expected, the more features used, the less likely that they align between the old and new image. Thus, the decoder knows the image is an imposter. In addition, features such as peaks or peaks of the derivative are robust to most transformation. Finally, since these features occur within the image, a global database is not required to determine where the image specific watermarks occur.

There may be a problem with backwards compatibility, meaning how does the detector know if the image has been tampered or the image is an old image made before the peak noise layer was added. There are three suggestions described below. The first suggestion is that a different group of global PN sequences could be used in this new version than with earlier versions. The second suggestion is to add a layer of noise

defining the version. The third is to use different spacing or position in the grid used to determine scaling and rotation of the embedded data.

In addition, when trying to find a watermarked area of a large image, feature-based watermarking is advantageous. As well known, searching the whole image for the small watermark is slow.

As shown in figures 1a and 1b, the process is to use a feature of the picture, such as the peak of the derivate, to embed a space-limited data, such as a local PN sequence, that provides information about the location of the picture's corner and the scaling. In addition, the whole block structure of the watermark, such as P by Q pixel areas for embedding (e.g., P and Q are preferably the same and multiples of two), could be based around this feature; thus, the feature-based watermark and embedded data carrying the message do not overlap. Using the peak of the derivative is ideal since the eye does not perceive noise near edges and it is robust to scaling and scanning. It is also efficient to find in the decoding process since only a few occurrences of the features should exist in the rest of the image. Finally, it is advantageous if the feature is not on the edge of the embedded area. If the feature is near an edge some embedded data, i.e. PN sequence, will be lost.

This embedded local-feature PN sequence will intrinsically inform the decoder that the feature is part of the picture by its existence. This local-feature PN sequence should also include a grid layer so that once it is found the scaling coefficient can be determined. Instead of a grid layer, the watermark decoder could employ the autocorrelation and cross-correlation scaling methods for compensating for scaling and rotation discussed in this document. This local-feature PN sequence should also include a few layers to provide where the lower-left (or other corner) of the picture is located. For example, two layers could inform the decoder which quadrant the feature was located. With the scaling and quadrant information, finding the global PN sequence, which carries the message, will be easier and faster.

Scaling

This method is illustrated through the following two embodiments. In the first embodiment, auto-correlation of an image and self-similar noise layer is used to determine the image's scaling and rotation.

Figure 2 shows the self-similar noise array layer that can be embedded within an image, or sequentially within audio, to determine the time scaling and rotation, for 2D images only. The PN variable is, for example, a 10x10 array of noise, where each PN sequence is identical. The 0 variable is, for example, a 10x10 array of zeros. There is a tradeoff between larger PN and 0 array sizes, which are less likely to be visible, and computations for autocorrelation. For example, when using 10x10 arrays, the autocorrelation only needs to include 20 multiply and add instructions per pixel to catch 0.5 to 2X changes.

The second embodiment includes estimating the image transformation by cross-correlating an original PN noise layer with an image which previously had the PN noise layer added and has been modified. Assuming the image has only been linearly transformed, such as by rotation or scaling, the PN noise layer is white, and the PN noise layer is orthogonal to the image, the result of the cross-correlation is the impulse function of the transformation. This impulse function can be used to improve recovery of the watermark. Finally, concepts from spectral estimation can be applied to increase the accuracy of the estimation since the assumptions are usually only partially true.

Transitions

In audio applications, the transition between embedding a 0 and 1 bit of auxiliary information occur by immediately changing the phase of the PN sequence, i.e. switch from multiplying by –1 and 1 and visa-versa. For example, after representing a 0 auxiliary bit by subtracting 100 ms of shaped noise from the signal, the 1 auxiliary bit is represented by adding the shaped noise to the next signal sample and so-on for 100 ms more. This is true in video applications. However, the eyes and ears are very susceptible to changes.

Thus, as shown in Figure 3, the transition between 0 and 1 bit of auxiliary information should have a transition period where the phase of the noise sequence is slowly changed. Although this will lower the embedded bit rate, it should decrease the perception of the watermark. The transition period length could be from 1 to 1several hundreds of a milliseconds.

Autocorrelation Watermarks

In general, a problem with reading watermarks via digital cameras, such as CCD or CMOS based cameras, is that the cameras integrate over space to get a color value. This integration is used since each camera receiving-element, such as a CCD, takes up space and a RGB or CMYK color grid is used. This integration does not degrade the picture quality since real-world pictures have data points that are correlated to its neighbor. However, with white noise-based watermarks, where the value changes every pixel, the camera not only removes the noise but also produces incorrect data since every pixel is independent in white noise. A current solution is to use noise where the value changes in blocks of pixels.

An alternative solution uses an autocorrelation based watermark, defined as taking a copy of the image, lowering its level, and placing it slightly offset from the original image. Either the offset value or copy level can be used to transfer 0's and 1's. For example, up and left shifts represent 1's, whereas down and right shifts represent 0's. The watermark is retrieved by calculating the autocorrelation function and finding the offset value of the peak, which is provided by the embedded low-level and shifted copy of the image.

This type of watermark survives integration since, as with real-world data, the neighboring will be related to each other and survive the camera's integration. This watermark will also be invisible since it intrinsically places the data where it can be hidden. In other words, an offset copy of the image is already prepared to be hidden in the image.

The prior-art shows this type of mark being used in audio, and bits are embedded sequentially, such as with US Patent #5,940,135 August 17, 1999 assigned to Aris Technologies, Inc, and included herein by reference. However, this process can only work with images in video. Thus, for single images, if the whole image is used, only one bit per image could easily be embedded and retrieved.

As shown in Figure 4a, a process that uses several blocks per image can be used to increase the embedded data rate. The block size is a balance between the number of embedded bits versus amount of noise embedded to retrieve one bit. In addition, the smaller the block size, more information is lost in edge patterns. Finally, the shift used in embedding the low level copy of the image should be minimal so as not to degrade

quality, such as blurring the edges. It appears desirable to have the shift larger than a single cameral pixel element, i.e. one CCD grid.

Finally, when splitting the image into blocks, the orientation of the blocks relative to the retrieved image is required. Traditionally, a noise grid covering each block is used. However, skipping the embedding process in some blocks can be used to locate the center or similar section of the image. In figure 4b, the X blocks contain watermarks, and the blocks without X's do not contain watermarks. As one can see, the non-watermarked blocks point to the center of the image as well as determine is rotation since they are asymmetrical.

Dynamic Media Scrambling

The problem with encrypting or scrambling content files is that they will be stored, such as on a hard-drive or optical disk, for a long time, possibly more than 10 years. This gives a pirate a long time to break the protection. As compared to other encrypted transactions, such as a bank withdrawal, if the pirate cannot break the code during the transaction, it is too late since the next transaction uses new keys. The current solution is to reject broken keys. However, this means that a legitimate user could find his/her content does not play and needs to be re-encrypted, or his/her device needs a firmware upgrade when he/she has done nothing. This will confuse and upset the customer.

The dynamic media scrambling process is to re-encrypt or re-scramble the content using a new technique or key each time the content is rendered (assuming the device is re-writeable), or using some other interval, possibly regular or not. This technique is invisible to the consumer. In addition, when keys are found to be broken, the removal of that key from the system will happen over time without any inconvenience to the legitimate consumer.

When content is rendered on the user's machine, the encryption routine decrypts the content using the current key. Then a new key is created, and the encryption routine encrypts the content for storage on the user's machine. To generate a new key, the encryption routine changes part or all of the previous key. In particular, part of the key may be based on something unique to the machine or software running on the machine, such as a processor ID, or date of the trash can or recycle bin in the operating

system. The remainder of the key changes with each rendering according to a random or pseudorandom function. When the new key is created, it is stored in a secure, encrypted and tamper resistant file on the user's machine. This key is used the next time the content is rendered.

The key not be changed each time the content is rendered. Alternatively, it may be changed each Nth time that the content is rendered, where N is some pre-determined integer. Alternatively, the key may be changed based on some external event trigger, such as the receipt of a new key from a local or remote key management system, or the receipt of a key update flag from a key management system or registry database that instructs the encryption routine on the user's device to update the key the next time the content is rendered.

This process of key updating enables encryption keys to be updated over time, and eventually move old or broken keys out of the system.

Transmarking

In many applications, a digital watermark signal embedded in media signals like audio, video and still images can be changed when the signal is transformed. Transmarking of the digital watermark may be used to change the embedded digital watermark technique at signal transformation points to be compatible with the new signal.

For example, when playing DVD audio over the radio, analog or digital radio, the watermark can be retrieved and re-embedded at a higher level or using a different technique at the broadcast location. Additionally, the watermark could be modified at a repeater station due to the increased noise level in the signal. This way an audio application can retrieve the watermark, while the original DVD can have the lowest change in perception due to the watermark as possible. More specifically, the audio application may be retrieving the watermark in a noisy room and artist won't complain that the DVD watermark ruins their recording.

This is a continuation of the subject matter in US Patent Application, Serial Number 09/404292, by Ken Levy and assigned to AIPL, filed on 9/23/99, which was

based upon Provisional Applications Serial Numbers 60/101,851 and 60/110,683 filed by Ken Levy on 9/25/98 and 12/2/98, respectively, all included herein by reference. These patent applications discussed changing the watermark type when audio was converted from raw PCM format to a compressed, such as MP3, AAC, Real, Liquid or other similar format.

This method also applies to video signals. For example, when watermarked DVD video is transferred to low bandwidth Internet video, such as provided by Real Networks, the DVD watermark is read and increased in amplitude or re-embedded to survive the massive compression needed to stream video over low bandwidth. This watermark may be used for copy protection, but could also be used to enable links or information about the video.

In some applications, it may be useful to convert auxiliary information embedded in a media signal from one format to another. This converting process is another application of transmarking. Transmarking may include converting an out of band identifier like a tag in a header/footer to a watermark or vice versa. It may also involve converting a message in one watermark format to another. The process involves a decoding operating on an input media object, and an encoding of the decoded information into the media object. It may also involve a process for removing the mark originally in the input object to avoid interference with the newly inserted mark.

There are a variety of reasons to perform transmarking. One is to make the embedded information more robust to the types of processing that the media object is likely to encounter, such as converting from one watermark used in packaged media to another watermark used in compressed, and electronically distributed media, or a watermark used in radio or wireless phone broadcast transmission applications.

This type of transmarking process may be performed at various stages of a media object's distribution path. An identifier in a watermark or file header/footer may be encoded at the time of packaging the content for distribution, either in an electronic distribution format or a physical packaged medium, such as an optical disk or magnetic memory device. At some point, the media signal may be converted from one format to another. This format conversion stage is an opportunity to perform transmarking that is

tailored for the new format in terms of robustness and perceptibility concerns. The new format may be a broadcast format such as digital radio broadcast, or AM or FM radio broadcast. In this case, the identifier may be transmarked into a watermark or other metadata format that is robust for broadcast applications. The new format may be a compressed file format (e.g., ripping from an optical disk to an MP3 format). In this case, the identifier may be transmarked into a file header/footer or watermark format that is robust and compatible with the compressed file format.

The transmarking process may leave an existing embedded identifier in tact and layer an additional identifier into the media object. This may include encoding a new watermark that does not interfere with an existing watermark (e.g., insert the new watermark in unmarked portions of the media object or in a non-interfering transform domain). It may also include adding additional or new identifier tags to headers or footers in the file format.

Fig. 5 is a flow diagram illustrating a process of transmarking. The input to the transmarking process is a digitally watermarked signal 20, such as an audio signal (e.g., a music track), a video signal, or still image. The digital watermark carries a message payload of one or more symbols (e.g., binary or M-ary symbols) conveying information such as a content identifier, transaction identifier, database index, usage or copy control parameters (flags instructing a device or process not to copy, copy once, not to transfer, etc.). There are a variety of applications for digital watermarks in multimedia content, including forensic tracking, broadcast monitoring, copy control, and using the watermark as a trigger for or link to interactive content to be rendered along with the watermarked signal, either in response to user input or automatically as the watermarked signal is playing.. Some of these applications are discussed in co-pending patent applications 09/571,422, 09/563,664, and 09/574,726, and 09/597,209 which are hereby incorporated by reference.

In these applications, there are a number of reasons to transmark the watermark signal embedded in a host signal. Some examples include: to increase the robustness of the watermark as it undergoes a format change (such as for compression, transmission, digital to analog conversion, up-sampling or down-sampling, printing, display, etc.), to reduce the perceptibility of the watermark before playback, or to

balance the trade-off of perceptibility levels vs. robustness levels of the watermark signal for a new as the host signal undergoes a change from one format to another.

The transmarking process illustrated in Fig. 5 begins by detecting a first watermark in the watermarked signal (22). A watermark detector employs a watermark key to identify the presence of a watermark. The specific operation of the detector depends on the watermarking process employed. In many techniques, the watermark key specifies the spatial, time, and/or frequency domain location of the watermark signal. It may also specify how to decode a message that has been modulated with a pseudo-random number (e.g., frequency or phase hopping, spread spectrum modulation). To simplify the search for the watermark, the watermark detector searches for reference signal attributes of the embedded signal, such as a known sequence of embedded symbols, or a known signal pattern in a particular time, space, or transform domain. These attributes enable the detector to determine whether a watermark is present in a suspect signal, and to determine its position within the time, space and/or transform domain.

Next, the watermark detector may optionally decode an embedded message (26), such as copy control parameters, content identifiers, owner identifiers, transaction identifiers, etc. This step is optional because the initial detection operation may convey enough information to trigger the remainder of the transmarking operation. For example, the mere detection of the presence of a watermark signal at a particular time, space, or transform domain location may convey one or more bits of message information.

Some examples will help illustrate the detection and message decoding process. One type of watermark embedding process encodes symbols by inserting scaled-amplitude, shifted versions of the host signal. The shift may be a combination of time, frequency, and/or spatial shifts of the host signal depending on the nature of the signal (e.g., time-frequency for audio, spatial frequency for imagery). This shifted version conveys message symbol values by the presence or absence of the shifted version or versions at a particular shift relative to the host, and/or by the amount of change effected to a statistical characteristic of the host signal by the embedding of the shifted version. Another type of embedding process embeds a watermark by modulating

perceptual domain samples and/or transform domain frequency coefficients. In both cases, the message may be randomized by applying a pseudo randomizing process (e.g., spreading a message by multiplying or XORing with a PN sequence) before making the changes to the host to hide the resulting message sequence in the host signal. The message may be embedded by an additive process of a modifying signal and/or by a quantization of sample values, frequency coefficient values, or statistical characteristic values.

In these embedding techniques, the detector looks for attributes of the watermark signal, such as by using correlation or a statistical analysis to detect the shifted versions or modulated samples/coefficients. By identifying evidence of known symbols or watermark signal attributes, the detector determines whether a watermark signal is present. In some cases, the watermark detector determines that an additional message payload message is present based on the detection of certain watermark signal attributes. It then proceeds to decode additional signal attributes and map them to message symbols. Further error correction decoding may be employed, such as BCH, turbo, Reed Solomon, and convolution decoding, to extract the message payload.

Next, the transmarking process removes the first watermark signal (28). Again, this process is optional because the transmarking process may proceed by embedding a second watermark without specifically attempting to remove or mitigate the effects of the first. Once the watermark detector has detected the watermark and determined its temporal, spatial, and/or frequency domain position, it can remove the watermark or mitigate its effect. It can substantially remove the watermark in cases where the embedding function is invertable, such as a reversible addition operation, by performing the inverse of the embedding function using the watermarking key to specify the attributes and location of the first watermark. It can also remove the watermark without knowing the inverse function, such as using a whitening filter with PN sequence based watermarking.

Interestingly, this could allow a less perceptible watermark to be added to content that is going from a low quality medium to a higher quality medium. Although the content will still be the quality of the original medium, the watermark will produce minimal or no further quality degradation. When transforming from high quality to

lower quality medium, removing the first watermark still improves quality and robustness due to reducing interference between each watermark.

In some applications, the watermarked signal may be converted to another format, such as compressing the signal before the transmarking process proceeds. These applications are ones where the signal in the new format is available for watermarking. In this case, the transmarking process proceeds by embedding a second watermark into the host signal after the format change has occurred. This enables the watermark embedding process to adapt the watermark to the perceptual quality and robustness parameters of the signal in the new format. In other applications, such as where the signal is broadcast, it is difficult or not practically possible to intercept the signal for embedding a new watermark after the format change occurs. For example, the format change may occur as a result of the broadcast transmission. In this case, the transmarking process proceeds to embed a second watermark and adapts the watermark to the robustness and perceptual quality parameters appropriate for the new format of the signal before the format change occurs.

Next, the transmarking process encodes the second watermark (44) using the same or some different embedding process as the first watermark (30). This second watermark can be added before the transformation, after the transformation, or during the transformation with a feedback loop. For example, the first watermark may be embedded by adding a shifted version of the host signal, while the second watermark may be embedded by adding a perceptually adapted pseudo random carrier signal in the perceptual or some transform domain (like Fourier, DCT, wavelet, etc.), or vice versa. The second watermark may modify different temporal, spatial or frequency portions of the host signal than the first, or the two watermarks may overlap in one or more of these portions of the signal. Regardless of whether the watermark embedding function is fundamentally the same or different as the one used to embed the first watermark, this embedding process (30) is specifically adapted to the perceptibility and robustness constraints of the new format or environment. This watermark embedding process uses robustness parameters (32) (e.g., watermark signal gain, extent of redundancy, frequency domain locations) to specify the watermark strength, redundancy and frequency domain locations that particularly adapt the watermark for survival in the

new format. This second watermark may add new information about the transformation where this information can be used for forensic tracking. The information could include any combination of the following: an identifier of the transformation device (such as an MPEG encoder device or manufacturer), and an identification of the distribution system, such as an identifier of the broadcast network or cable system. This new information augments the original information embedded into the first watermark and does not alter its meaning, but instead, adds additional payload information.

To ensure that the second watermark satisfies robustness constraints, the embedding process optionally applies a feedback path that applies the watermarked signal to a degradation process, then measures the number of errors incurred in decoding a known message, and selectively increases the gain of the watermark signal in the portions (temporal, spatial or frequency portions) of the signal where the errors occurred. The degradation operations may include a compression operation, or an operation that models degradation likely to be encountered in the new format, such as digital to analog conversion, printing/scanning, broadcast transmission, time scale changes, etc. This process repeats until the measured error rate falls below an acceptable threshold.

In addition, the embedding process uses perceptual quality parameters 33 that specify constraints on perceptual quality of the signal for the new format. These parameters may specify limits on the watermark strength, or define a perceptibility threshold that can be measured automatically, like Peak Signal to Noise Ratio, typically used in analysis of digital watermarking methods. Again, as above, the embedding process optionally includes a feedback path that measures the perceptual quality of the watermarked signal and selectively reduces the gain of the signal in the portions of the signal (temporal, spatial or frequency portions) where the watermarked signal exceeds the perceptibility threshold.

Fig. 5 graphically depicts the interaction between the watermark embedding process 30, on the one hand, and the rendering/editing environment or transmission environments (34, 36) on the other. This diagram depicts how the embedder adapts the new watermark to the environment in which the transmarked signal will be used. For

example, if the signal is a still image that is being used in a photo editing software environment, the robustness of the watermark can be adapted to the image processing operations in the editing tool. If the watermark is going to need to survive printing, then the transmarking process embeds the signal with a new watermark designed to survive that process and be recoverable via an image scanned from the printed image. In this case, the watermark embedder may include additional calibration signal information as set forth in US Patent 5,862,260 to ensure that the watermark can be detected despite geometric distortion.

As an aside, just as the second watermark may be adapted to the intended environment, the operations in the editing tool can be modified so as to improve the survivability of the watermark. In this case, the image editing operations such as blurring, color transformation, etc. are adapted to preserve the watermark signal to the extent possible. In particular, a low pass filter or blur operation that typically reduces high frequency components may be implemented so as to pass selected high frequency components to maintain the watermark signal in those components. The operation of adding guassian noise may be modified by shaping or reducing the noise at certain frequencies to reduce interference with the watermark signal at those frequencies. In cases where watermarks are inserted by modifying a particular color channel such as luminance, the color transform operations may be designed to preserve the luminance of the watermarked image.

Further, the signal editing tool may be integrated with the transmarking process to decode the watermark before an operation, and then re-encode the watermark after an operation to ensure that it is preserved. For example, the wateramark may be re-applied after the image editing tool is used to make an affine transform of an image, or after the image is cropped.

In the case of transmission of media signals over a communication channel, the watermark may be transmarked at points in the communication channel where the signal (audio, video, or image signal) is transformed. These include cases where the signal is un-compressed and re-compressed in another format, where the signal is transformed in a router or repeater (e.g., when the signal is amplified in a router or repeater node in the communication path, the watermark is transmarked at higher

intensity), where the signal is transformed into packets in a switching network, the watermark signal may be decoded and re-encoded in the individual packets, or re-encoded after the signal is re-combined. The re-encoding is effected by transferring a watermarking command in the header of the packets specifying the watermark payload and watermark embedding protocol to be used in the re-combined signal.

In audio and video compression codecs, the transmarking process may be integrated into the compression codec. This enables the codec to modify the compression operation or modify the bitrate to ensure that the watermark survives. In the first case, the compression codec may be designed to preserve certain frequency components that would otherwise be substantially reduced to preserve the watermark. In the latter case, the codec selects a bit rate at which the watermark survives, yet the signal has been compressed to an acceptable level.

If the watermarked signal is going to be rendered in a high fidelity device where usage is tightly controlled, such as a DVD player, the second watermark can be embedded so as to have less impact on perceptibility. Conversely, if the watermarked signal is going to be rendered in a lower fidelity device, such as a personal computer, the second watermark can be embedded so that it is more robust while staying within the perceptual quality parameters of the rendering device. In addition, the watermark can be changed if DVD audio masters are converted to CDs or cassette tapes.

If the watermarked signal is going to be transmitted, such as in the broadcast environment, the embedding process encodes the second watermark with robustness to survive the broadcast and maintain the perceptual fidelity within the less rigid constraints of the broadcast environment. The transmarking process can be used to encode triggers used in interactive video or audio. The triggers may be originally encoded in one format and transmarked into another format before broadcast, or at some node in the broadcast process. For example, the trigger can transmarked in video when it is compressed into MPEG2 format for broadcast, or when the content is received at a cable head-end or node in the content distribution channel. The trigger may be a network address of interactive content like an IP address or URL, or an index to a network address, interactive content like HTML or XML, etc.

As another example, triggers for interactive content in radio broadcasts can be transmarked when the content is transferred from a packaged medium, such as an optical disk, and prepared for broadcast over traditional radio broadcast, digital satellite broadcast, or Internet streaming broadcast.

Like the first watermark, this second watermark employs a watermarking key 38 to specify the spatial, time and or frequency attributes of the second watermark. In addition, the message decoded from the first watermark, such as an identifiers 40, copy control parameters 42 are embedded.

The result of the transmarking process, in a typical case, is a new watermarked signal 46. As noted, the information or function of the watermark may be transmarked to out-of-band data like a file header or footer, such as an ID3 tag in MP3 audio. Conversely, out-of-band data may be transmarked into in-band data that is embedded into the host signal using a digital watermarking process.

Watermark Embedding Functions in Rendering Description Files

Document and other media object generation tools continue to increase in sophistication and complexity. Adobe offers a variety of such tools, including their InDesign software. Watermarking can advantageously be effected in such systems.

In such environments, a document may be created using a variety of tools – most of which can insert a watermark. One program may use as input the output of one or more other programs (i.e., "compositing").

To better handle watermarking in this environment, a watermarking function (e.g., a PostScript-like command) can be provided in the tools. This function is called with parameters specifying the desired features of the watermark information, e.g., payload, robustness level, masks to be used. At rendering time, such as for on-screen viewing, printing proofs, or ripping the final version, the watermark is actually added as digital data. In such environment, the embedder knows the properties of the rendering device, such as the printer, and appropriately adjust its embedding accordingly. With this concept, watermarks are not lost during composite operations, and watermarks can be embedded in vector (or line) art. Moreover, the color manager at the ripping stage may be the best entity to add the watermark.

This idea likewise extends to video - especially MPEG-4 object video, audio - especially MIDI or MPEG-4 structured audio language, and virtual advertisements.

The use of a PostScript-like function to embed a watermark is further detailed in application 09/629,401.

An alternate method is that no desktop tool has watermarking capability, but instead an on-line watermarking server is available to support common image formats. A variety of tools are enabled to submit images to the server with information regarding the desired parameters of the watermark. The server then returns the image to the application. In this way, the burden of integration is virtually eliminated and the registration and marking take place simultaneously.

When watermarking graphic art material, such as packaging, it is desirable to have the graphic designer, rather than the printer, embed the desired watermark for the person or company creating the packing. Having the graphic artist embed the watermark is advantageous because the consumer is already communicating with the artist, and the customer may never need to communicate with the printer. Usually printers are only needed to proof the plates or prototype. In addition, printers don't want extra things to remember, printing is hard enough.

However, much graphic art material remains as line-art (also known as vector graphics) until being rasterized during the printing process, and the state of the art for watermarking is raster based.

A solution is to embed watermark functions in the line-art file, in a similar fashion to how fonts are described with a Bezier curve. The watermark function contains the bits to embed as well as rules how to embed these bits in different elements. The watermark function could be considered as a command in the popular expanded postscript (EPS) format.

For example, when producing text and a watermark is contained in the line-art, the watermark bits could be embedded by slightly adjusting the position, either vertical, horizontal or both, of each letter. Alternatively, the watermark could be embedded by adding or removing bumps, which are too small to see but can be read digitally, on the edges of the letters. Importantly, any data embedding method can be used according to the bits and rules of the watermark function. Similarly, when producing drawing

objects, the watermark function could be implemented by embedding the bits in bumps along the edges of the object. Alternatively, when putting a gradient fill inside an object, the watermark function could be implemented by adding more traditional PN sequences within the gradient fill, or modulating halftone dots.

In general, the watermark function contains the bits to be embedded and rules or links to how to embed these bits. Thus, the watermark function is implemented according the desired embedding method when the line-art is rendered, such as on the screen, printer or printing plates.

As noted, the watermarking function may be applied in variety of types of media objects and rendering description languages. Figs. 6 and 7 illustrate a framework for implementing and using the watermark embedding function as a rendering command. Fig. 6 is a diagram illustrating a watermark embedding command (100) and insertion of the command into a rendering description file (102). The watermark embedding command is specified in a text format or some other binary form compatible with the rendering description file in which it is inserted.

At the time of media signal creation, the user specifies the watermark embedding command and associated parameters. Later, at the time of rendering, the rendering device invokes a watermark embedding module to embed the watermark in the media object according to the watermark embedding command. The watermark command parameters include a combination of parameters describing the watermark message payload, the watermark protocol, the watermark embedding method, the payload specification, the embedding locations, the robustness parameters, and the perceptual quality parameters. Any combination of these and other parameters may be used depending on the application.

The watermark message comprises some number of binary or M-ary symbols. These symbols can represent a variety of types of information related to the media signal in which they are embedded, including, to name a few:

- copy control parameters controlling rendering or transfer of the object,
- identifiers of the media object, its owner, or transactions of the object (user ID, machine ID, storage device ID, etc.),
- network addresses of related information, programs, web sites, etc.

- program or device instructions
- metadata for the media object
- an index (or several indices) to a database entry relating to the object that stores the above information or other information such as programs that are executed in response to watermark detection, etc.

The watermark protocol specifies how the watermark message is to be embedded and the meaning of the various symbols in the watermark message. The protocol may be specified using one or more parameters. These protocol parameters include a parameter that specifies the embedding method, such as a pointer to a embedder module or plug-in to be used in the rendering device to embed the watermark. There are several different embedding methods per media type. For image signals including video and still images, the method may include a spatial or frequency domain spread spectrum watermark embedder, a watermark embedder that encodes symbols by adjusting samples or features to quantization levels associated with symbols to be embedded, halftone modulation methods (varying halftone dot shapes, screens, error diffusion thresholds, dot cluster sizes or widths according to changes associated with message symbols, etc.). For audio signals, the method may include temporal or frequency domain spread spectrum watermark embedder, a watermark embedder that encodes symbols by adjusting samples or features to quantization levels associated with symbols to be embedded, a watermark embedder that encodes a collection of masked tones or time/frequency shifted versions of the host signal corresponding to symbols to be embedded, etc. In some cases, the method may be left unspecified so that the rendering device or transmission channel may optimize the watermark method and protocol for that rendering device or channel. In this case, the rendering device or channel has a compatible decoder associated with that device or channel for decoding the watermark. Alternatively, a universal watermark signal or metadata may be used to specify the watermark type for decoding.

The protocol parameters may also include more detailed information about the watermark payload, namely a payload specification. The payload specification may includes items such as the type of error correcting codes to employ, the type of error

detection to employ, the number of message symbols (e.g., binary bits) in the payload, encryption keys for encrypting the payload, etc.

The protocol may also specify where to embed the watermark, which is referred to as the "embedding locations" in Fig. 6. The embedding locations include, and are not limited to, spatial, temporal, and transform domain locations to embed the watermark in the host media signal. The transform domain locations refer to transform domain coefficients or sets of coefficients in particular block size of content. Examples of transform domains include Fourier domain, wavelet domain, DCT, etc. The embedding locations may specify, for example, that the watermark is to be confined to certain frequency ranges in the signal. Also, for images and video, the embedding location may also specify the color plane or planes in which to embed the watermark signal, such as the luminance channel, the blue channel, or some other color channel.

In some applications, the watermark embedder will embed different message payloads in different parts (spatial, temporal, frequency, transform domain portions) of the host media signal. In these cases, the watermark embedding command specifies the parameters for each of the different message payloads, including its embedding location, intensity, fragility (for fragile watermarks), robustness parameters, perceptual quality parameters, redundancy, etc. This enables the watermark embedder module (or modules) to embed combinations of different robust watermarks, robust and fragile watermarks, or fragile watermarks at varying degrees of fragility. In some cases, the message payload may be a single bit, which reduces to the presence or absence of a watermark signal. This single bit may be spread in a signal covering several embedding locations, repeated in several instances of the same signal, or some combination of both.

As noted previously, the embedding locations may be specified in terms of spatial, temporal or transform domain masks that specify the areas for embedding the watermark. The mask is an array of elements each corresponding to an embedding location. For each element, the mask may be associated with other parameters, such as the payload for that location, the robustness for that location, and the perceptual quality for that location. The mask may be designed by the creator of the media object to

specify where to, and conversely, where not to embed the watermark, and also to specify the watermark intensity for the areas where it will be embedded.

The robustness and perceptual quality parameters enable the user or application that inserts the embedding command to control the trade-offs between robustness of the watermark and perceptibility. The robustness parameters may be specified in terms of intensity (e.g., watermark signal gain for a particular embedding location), redundancy (e.g., the extent to which the message payload is redundantly encoded across embedding locations to increase its robustness), and frequency locations (e.g., the extent to which the watermark signal is concentrated in lower frequency areas that are more likely to survive transformations of the host signal). Each of these parameters may be specified as a preferred range to enable the embedding module to optimize the watermark for perceptibility and robustness in specified robustness and perceptibility ranges.

Related to the robustness parameter, the watermark embedding command may also specify the level of fragility of the watermark at particular locations in the media signal. Such fragile watermarks are embedded in response to the embedding command. Later at watermark decoding, the presence of the fragile watermark, or its measured strength (e.g., as measured by the error detection rate of a known embedded symbol set, or by threshold levels of detected watermark strength), are used to detect tampering or processing of the watermarked signal.

This type of robustness and perceptual quality specification enables the watermark embedder module to perform iterative embedding with a feedback path to optimize embedding for a particular rendering or transmission device. In this iterative approach, the embedder initially embeds the watermark payload according to the command parameters at lowest robustness and highest perceptual quality, applies a model of degradation for the particular rendering device or transmission channel to the watermarked signal, and then decodes the watermark to measure the detection error rate for the message payload (e.g., the detection error is quantified using a measure of the difference between decoded symbols and expected symbols before error correction decoding is applied). It then repeats another iteration of this process, increasing the robustness slightly with each iteration until the detection error rate is at a satisfactory

level. The model of the degradation may be a compression operation, or a signal transformation that simulates the distortion due to digital to analog – and analog to digital conversion, time scaling, affine transformation, etc.

The perceptual quality parameters may be specified using automated measures such as peak signal to noise ratio, which quantifies the distortion of the watermarked signal relative to the un-watermarked signal. The perceptual quality parameter may be specified as an allowable range or as a threshold which should not be exceeded.

A media object creation program inserts the watermark embedding command into the rendering description file 102 as another rendering command. As shown in Fig. 6, the rendering description file includes a collection of rendering commands (104, 106, 108) that reference media signals (110, 112) or descriptions of media signals (e.g., 114, such as the case for vector graphics file) to which the rendering commands are to be applied. This file may then be stored for later use, sent to a rendering device, or distributed over a transmission channel.

There are a variety of potential formats for the rendering description file, such as postscript, PCL, EPS, PDF, job tickets, vector graphics, etc. for images and documents, structured audio and MIDI for audio, and MPEG-4 or MPEG-7 for video and audio.

Fig. 7 is a process for embedding watermarks in media objects using watermark embedding commands. The process begins when a user or application program inserts the watermark embedding function as a rendering command (120) into the rendering description file (122). Later, when the media object described in the rendering description file is prepared for rendering, the rendering process (124, 126, 128) reads the watermark embedding command, and invokes the appropriate watermark embedding module (e.g., 130, 132) to embed the watermark according to the parameters specified in the embedding command (120). The watermark embedding module is adapted for the particular rendering device (134, 136, 138) that will render the signal or the transmission channel (140) that will communicate the signal. To avoid degradation to the signal due to the transmission channel, it can be sent through the transmission channel as a rendering description file and later rendered and embedded with the watermark at the rendering device.

For images, the rendering process may be implemented in a display driver, printer driver, or plug-in to the display or printer driver. It may also be implemented in the printer hardware and specifically integrated into the halftoning process so that the watermark is particularly adapted to the halftone process and is embedded into the image after or while it is rasterized to a halftone image. This technique applies to a variety of halftone processes including ordered dithering (e.g., blue noise masks, clustered dot halftones, etc.), error diffusion, stochastic screening, etc. Examples of halftone watermark embedding methods include:

1. Adding a perceptually adapted spread spectrum watermark signal to an image in multi-level per pixel format at the halftone dot resolution before converting the image to a halftone image. The watermark signal is created by convolving or multiplying the message payload with a pseudorandom carrier signal, and then scaling the carrier signal based on the masking attributes of the image;

2. Modulating the error threshold used in error diffusion halftoning according to a perceptually adapted spread spectrum watermark signal,

3. Modulating line widths of halftone dots;

4. Modulating halftone cluster shapes and sizes to embed a watermark signal into a halftone image; or modulating halftone screens according to predetermined relationship between . For more information about watermark embedding methods for halftone images, see U.S. Patent Nos. 09/074,034, entitled Methods and Systems for Watermark Processing of Line Art Images, 09/689,226, entitled Halftone Watermarking and Related Applications, and 60/263,987, entitled Halftone Primitive Watermarking and Related Applications, which are hereby incorporated by reference.

For images, audio and video, the rendering process is implemented in media object generation tools used to transform the signal into a format for distribution, broadcast, or transmission. In these cases, the signal transformation process selects the embedding method and parameters that adapt the robustness of the embedded watermark and perceptual quality of the rendered watermarked signal for the particular rendering process or transmission channel. For example, an audio processor renders a music signal and embeds the watermark payload at a robustness level appropriate for the distribution, broadcast or transmission format. Similarly, a video processor renders

a video signal and embeds the watermark payload at a robustness level appropriate for the distribution, broadcast or transmission format.

The watermark function can specify that the watermark be embedded as part of the signal formatting process, such as part of the process of compressing the image, video or audio signal. This enables the watermark module to interact with the compression process to embed the watermark so that it is adapted to that format, e.g., embedding in the compressed data stream or partially compressed stream. The compression rate of the signal can be adaptively set by determining the greatest extent of compression where the watermarked signal still survives based on an error detection measure. Similarly, the perceptual quality parameters may be used to tune the compression process so that the compression rate is selected that maintains the desired perceptual quality of the signal and the robustness level of the watermark signal.

Alternatively, the watermark function can specify that the watermark be embedded after it is converted to a particular format for rendering or transmission (e.g., embedded after compression, or conversion to a broadcast format). The rendering or transmission channel provides robustness and perceptual quality parameters about that rendering process or transmission channel to the embedder module so that it can optimize the watermark embedding for the particular rendering process or transmission channel. In particular, it specifies the watermark robustness, e.g., intensity, or quality constraints that the watermark embedder must adhere to while embedding the payload specified in the watermark embedding command.

The watermark embedder module queries the rendering process, device or transmission channel for its robustness and perceptual quality attributes. If the quality requirements are lower, then the embedder can increase the robustness of the watermark within an allowable range specified by the watermark embedding command parameters. Conversely, if the quality requirements are higher, then the embedder can select the lowest allowable robustness level specified in the watermarking command to embed the watermark so as to minimize degradation to perceptual quality due to the watermark. The same process can be applied to tune the embedding operation based on the robustness attributes of the rendering process or transmission channel. If the rendering process is expected to substantially degrade the watermark's detectability,

then the embedder can select the most robust level for the watermark within the allowable range of the watermark embedding command.

Rather than querying the rendering device or channel, the watermark embedding command can be designed to select automatically the preferred watermark embedding method for that device or channel.

The watermark embedding function is particularly well suited for controlling the embedding of watermarks in vector graphics used in virtual advertising for streaming media, like streaming video. The virtual advertising is a vector graphic such as a logo that is superimposed on a video sequence when the streaming video is rendered in a receiving device, such as television equipped with a set top box or a personal computer on the Internet. This vector graphic file defining the virtual advertising can include a watermark embedding command as described above. At rendering time when the vector graphic is rendered, a watermark embedder module at the receiver embeds a watermark onto the vector graphic. This vector graphic can be used as a trigger for interactive TV applications wherever that video travels. For example, the user clicks on (or otherwise selects the logo displayed on the video screen with a cursor control device) to request interactive information such as a web page or to order a product or service when playing previously recorded or live content through a personal video recorder like a Tivo machine. The watermark in the logo is then decoded and a payload is extracted from it that indexes a database entry. The database returns the interactive information (URL, HTML, web page, etc.) or some other programmatic code that executes on the user's set-top box or computer and enables the user to buy the advertised product. As illustrated in this example, the watermark embedding command may be specified for content that includes a combination of different media signals like video, vector graphics, and audio, that get combined at rendering time in the receiving device.

Concluding Remarks

Having described and illustrated the principles of the technology with reference to specific implementations, it will be recognized that the technology can be implemented in many other, different, forms. To provide a comprehensive disclosure

without unduly lengthening the specification, applicants incorporate by reference the patents and patent applications referenced above.

The methods, processes, and systems described above may be implemented in hardware, software or a combination of hardware and software. For example, the auxiliary data encoding processes may be implemented in a programmable computer or a special purpose digital circuit. Similarly, auxiliary data decoding may be implemented in software, firmware, hardware, or combinations of software, firmware and hardware. The methods and processes described above may be implemented in programs executed from a system's memory (a computer readable medium, such as an electronic, optical or magnetic storage device).

The particular combinations of elements and features in the above-detailed embodiments are exemplary only; the interchanging and substitution of these teachings with other teachings in this and the incorporated-by-reference patents/applications are also contemplated.

We claim:

1. A feature based watermark embedding method for hiding auxiliary data in a media signal comprising:

identifying N features in the media signal, where N is an integer greater than one; and

embedding a watermark around the N features by modulating sample values in a group of samples around each feature according to a watermark signal layer.

2. The method of claim 1 wherein the features comprise peaks of a derivative of the media signal.

3. The method of claim 2 wherein the media signal is an image.

4. The method of claim 1 wherein the watermark signal is a noise layer.

5. The method of claim 1 wherein different types of watermark signals are embedded into the media signal: a first type embedded around the N largest features in the media signal, and a second type spread around the media signal.

6. The method of claim 5 wherein the media signal is an image.

7. The method of claim 6 wherein the features comprises peaks of the image.

8. The method of claim 7 wherein the features comprise peaks of the derivative of the image.

9. The method of claim 5 wherein the second type carries a message payload.

10. The method of claim 1 wherein the watermark signal has correlation properties that enable a watermark decoder to compensate for scaling distortion by

computing auto or cross correlation of the watermarked signal and deriving scaling from positioning of peaks in a resulting signal.

11. The method of claim 1 wherein the watermark comprises a PN sequence of symbol values of either 1 or –1, the symbols are mapped to samples of the media signal, and transitions in phase between adjacent groups of samples corresponding to different symbols are made to change slowly.

12. A method of steganographically embedding a watermark in a media signal comprising:

forming a message comprising a PN sequence of symbols of either 1 or –1;

mapping the symbols to samples in the media signal;

smoothly varying transitions in phase between adjacent groups of samples corresponding to different symbols.

13. The method of claim 12 wherein the media signal is an audio signal.

14. A method of decoding a feature based watermark that hides auxiliary data in a media signal comprising:

identifying N features in the media signal, where N is an integer greater than one; and

decoding a watermark around the N features by correlating sample values in a group of samples around each feature with a watermark signal layer;

using one of the N features as a reference for decoding auxiliary data for one or more of the other features.

15. The method of claim 14 wherein the features are peaks in the media signal.

16. The method of claim 15 wherein the peaks are peaks of a derivative of the media signal.

17. The method of claim 14 wherein correlation peaks generated by autocorrelation of the watermarked signal are used to compensate for scaling distortion of the watermarked signal before decoding an auxiliary data message.

18. The method of claim 17 wherein the media signal is an image and the correlation peaks are used to compensate for rotation distortion before decoding the auxiliary data message.

19. The method of claim 14 wherein multiple different watermark layers are decoded including watermark layers around the features and elsewhere.

20. The method of claim 19 wherein the watermark layers employ PN sequence carrier signals detected using cross correlation or autocorrelation functions on the watermarked signal.

21. A method of transmarking a media signal previously embedded with a first digital watermark using a first digital watermark embedding method, comprising:

detecting the first digital watermark in the media signal;

embedding message information from the first digital watermark into a second digital watermark in the media signal before the media signal undergoes a transformation process such that the second digital watermark is adapted to survive the transformation process.

22. The method of claim 21 wherein the second digital watermark is increased in amplitude relative to the first digital watermark to survive the transformation process.

23. The method of claim 21 wherein the second digital watermark is embedded using a different steganographic embedding method than the first digital watermark embedding method.

24. The method of claim 21 wherein the first digital watermark is at least partially removed before embedding the second digital watermark.

25. The method of claim 21 wherein the message information includes message symbols and further including:

decoding the message symbols from the first watermark; and

re-embedding the message symbols from the first watermark into the second watermark.

26. The method of claim 25 wherein the message symbols include an index to a database entry that stores information about the media signal.

27. The method of claim 25 wherein the message symbols include a content identifier.

28. The method of claim 21 wherein the second digital watermark is embedded using a robustness parameter that is used to control embedding so that the second digital watermark is adapted to survive the transformation process; and the robustness parameter is specified by a rendering, editing or transmission process that is going to process the media signal after the second digital watermark is embedded in the media signal such that the second digital watermark is adapted to robustness constraints of the rendering, editing or transmission process.

29. The method of claim 28 wherein the robustness parameter specifies watermark signal strength, redundancy, or frequency domain locations of the second digital watermark so that the second digital watermark is more likely to survive the transformation process than the first digital watermark.

30. The method of claim 21 wherein the second digital watermark is embedded using a perceptual quality parameter that is used to control embedding so that the second digital watermark has a perceptual quality adapted for the transformation

process; and wherein the perceptual quality parameter is specified by a rendering, editing or transmission process that is going to process the media signal after the second digital watermark is embedded in the media signal such that the second digital watermark is adapted to perceptual quality constraints of the rendering, editing or transmission process.

31. The method of claim 21 wherein the second digital watermark is embedded using a feedback process that repeatedly embeds at least portions of the second digital watermark and selectively adjusts the strength of the second digital watermark in portions of the media signal according to degradation of the watermark measured after applying a degradation process to the watermarked signal or according to perceptual quality measurements.

32. A computer readable medium on which is stored software for performing the method of claim 21.

33. A method of transmarking a media signal previously embedded with a first digital watermark using a first digital watermark embedding method, comprising:

detecting the first digital watermark in the media signal;

converting the media signal to a different format;

embedding message information from the first digital watermark into a second digital watermark in the converted media signal such that the second digital watermark is adapted to robustness or perceptibility parameters associated with the new format.

34. The method of claim 33 wherein the new format is a compressed format of the media signal.

35. The method of claim 33 wherein the second digital watermark is encoded with greater signal strength than the first digital watermark to survive transformation of the media signal in the new format.

36. The method of claim 33 wherein the second digital watermark is encoded with lesser signal strength than the firs digital watermark so as to be less perceptible in the new format of the media signal.

37. The method of claim 33 wherein at least a portion of the first digital watermark is removed before converting the media signal to the different format.

38. A computer readable medium having software for performing the method of claim 33.

39. A transmarker for transmarking a media signal previously embedded with a first digital watermark using a first digital watermark embedding method, comprising:

means for detecting the first digital watermark in the media signal;

means for embedding message information from the first digital watermark into a second digital watermark in the media signal before the media signal undergoes a transformation process such that the second digital watermark is adapted to survive the transformation process.

40. A transmarker for transmarking a media signal previously embedded with a first digital watermark using a first digital watermark embedding method, comprising:

means for detecting the first digital watermark in the media signal;

means for converting the media signal to a different format;

means for embedding message information from the first digital watermark into a second digital watermark in the converted media signal such that the second digital watermark is adapted to robustness or perceptibility parameters associated with the new format.

41. A method for controlling embedding of a digital watermark in a media object, comprising:

receiving a watermark embedding function specifying watermark embedding parameters, including watermark intensity and message payload;

inserting the watermark embedding function into a rendering description file; and

at rendering time, reading the watermark embedding function and steganographically embedding the watermark message payload into the media object at the watermark intensity.

42. The method of claim 41 wherein the media object comprises graphic art including a collection of two or more images in different formats.

43. The method of claim 41 wherein the media object comprises a music signal, and the steganographic embedding process is adapted to the robustness or perceptual quality parameters selected at the rendering time.

44. The method of claim 41 wherein the media object comprises a music signal, and the steganographic embedding process is adapted to the robustness or perceptual quality parameters selected at the rendering time.

45. The method of claim 41 wherein the steganographic embedding process includes:

iteratively embedding the message payload in two or more iterations;

analyzing the error detection rate of the message payload in each iteration; and

adjusting a robustness parameter of the embedding process for at least one of the iterations so that the error detection rate is at an acceptable level.

46. The method of claim 41 including:

providing two or more different watermark embedding modules, each adapted for different rendering processes or transmission channels, the watermark embedding modules being selected at rendering time depending on the rendering process or transmission channel to which the media object is to be applied.

47. The method of claim 41 wherein the watermark embedding function specifies the embedding locations of the watermark.

48. The method of claim 41 wherein, at rendering time, the steganographic embedding process selects the embedding locations of the watermark depending on the rendering process.

49. The method of claim 41 wherein the media object comprises an image, and the steganographic embedding process embeds the watermark message payload into the image after the image is rasterized into a format compatible with a printer on which the image is to be printed.

50. The method of claim 51 wherein the watermark message payload is embedded in the image after it is converted into a halftone image by a halftoning process compatible with the printer on which the image is to be printed.

51. A media object processing system comprising:

input means for enabling a user to specify rendering commands for a media object, including a watermark embedding function to be applied to the media object, the watermark embedding function including a watermark message payload, and parameters controlling embedding of the watermark message payload in the media object;

means for creating a rendering description file describing how to render the media object; and

a watermark embedder module for steganographically embedding the watermark message payload into the media object.

52. The system of claim 51 where in the system is operable to select different embedding modules for different rendering processes.

53. The system of claim 51 wherein the embedder module is operable to iteratively embed the watermark message payload in the media object in two or more iterations, and with each iteration, analyzing an error detection rate of the message payload to adapt robustness of the watermark.

54. The system of claim 53 wherein the watermark embedder module applies a degradation process to a watermarked media signal output from an iteration before analyzing the error detection rate.

55. The system of claim 51 wherein the media object comprises graphic art that is at least in part specified by a rendering command that has not be converted into a rasterized image when the watermark embedding function is inserted in the rendering description file.

56. The system of claim 51 wherein the media object comprises structured audio represented, at least in part, by a rendering command that has not been converted to an audio signal when the watermark embedding function is inserted in the rendering description file.

57. The system of claim 51 wherein the media object comprises rendering commands specifying how to create a video sequence that have not been converted to a video signal when the watermark embedding function is inserted in the rendering description file.

58. A computer readable medium having a rendering description file comprising:

one or more rendering commands describing how to render a media object; and

a watermark embedding function specifying how to embed a watermark message payload into the media object after the media object has been rendered.

59. The computer readable medium of claim 58 wherein the rendering description file includes a rendering command describing how to create a rasterized image, the watermark embedding function specifying how to embed the watermark message in the rasterized image.

5

60. The computer readable medium of claim 59 wherein the watermark embedding function including a watermark intensity parameter and embedding locations specifying where and at what intensity to embed the watermark message in the rasterized image.

10

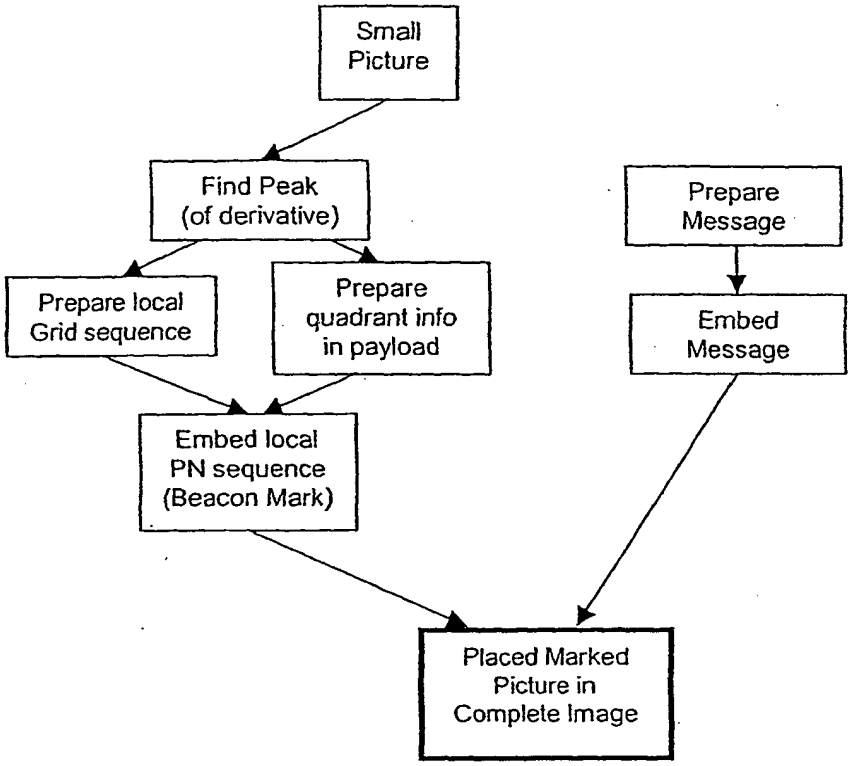

Small Picture
Find Peak (of derivative)
Prepare Message
Prepare local Grid sequence
Prepare quadrant info in payload
Embed Message
Embed local PN sequence (Beacon Mark)
Placed Marked Picture in Complete Image

Fig 1a

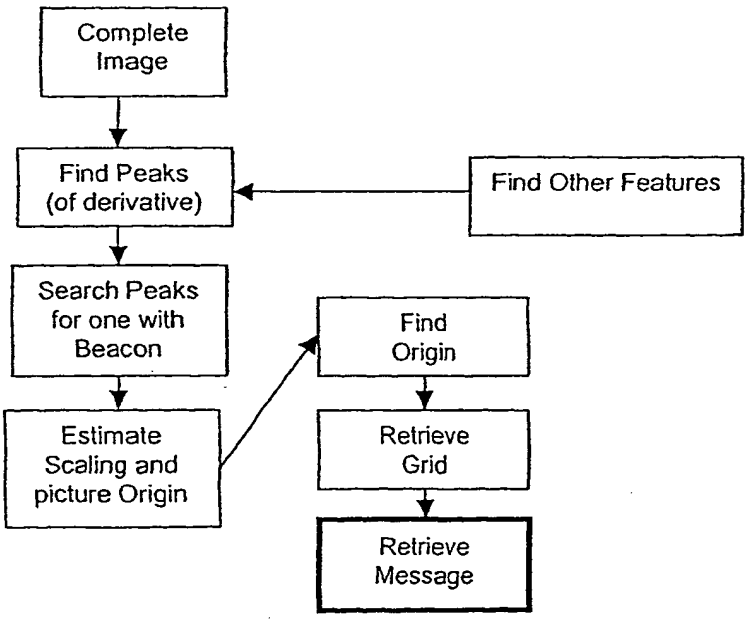

Complete Image
Find Peaks (of derivative)
Find Other Features
Search Peaks for one with Beacon
Find Origin
Estimate Scaling and picture Origin
Retrieve Grid
Retrieve Message

Fig 1b

PN	0	PN	0	PN	0
0	PN	0	PN	0	PN
PN	0	PN	0	PN	0
0	PN	0	PN	0	PN

Fig 2

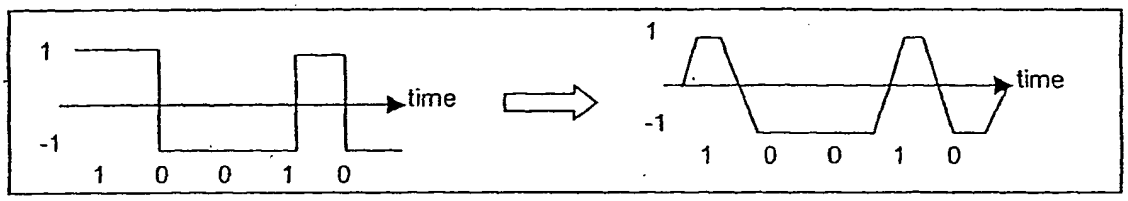


Fig 3

Fig 4a

X	X		X	X
X	X		X	X
X	X	X		X
X	X	X	X	

Fig 4b

Fig. 5

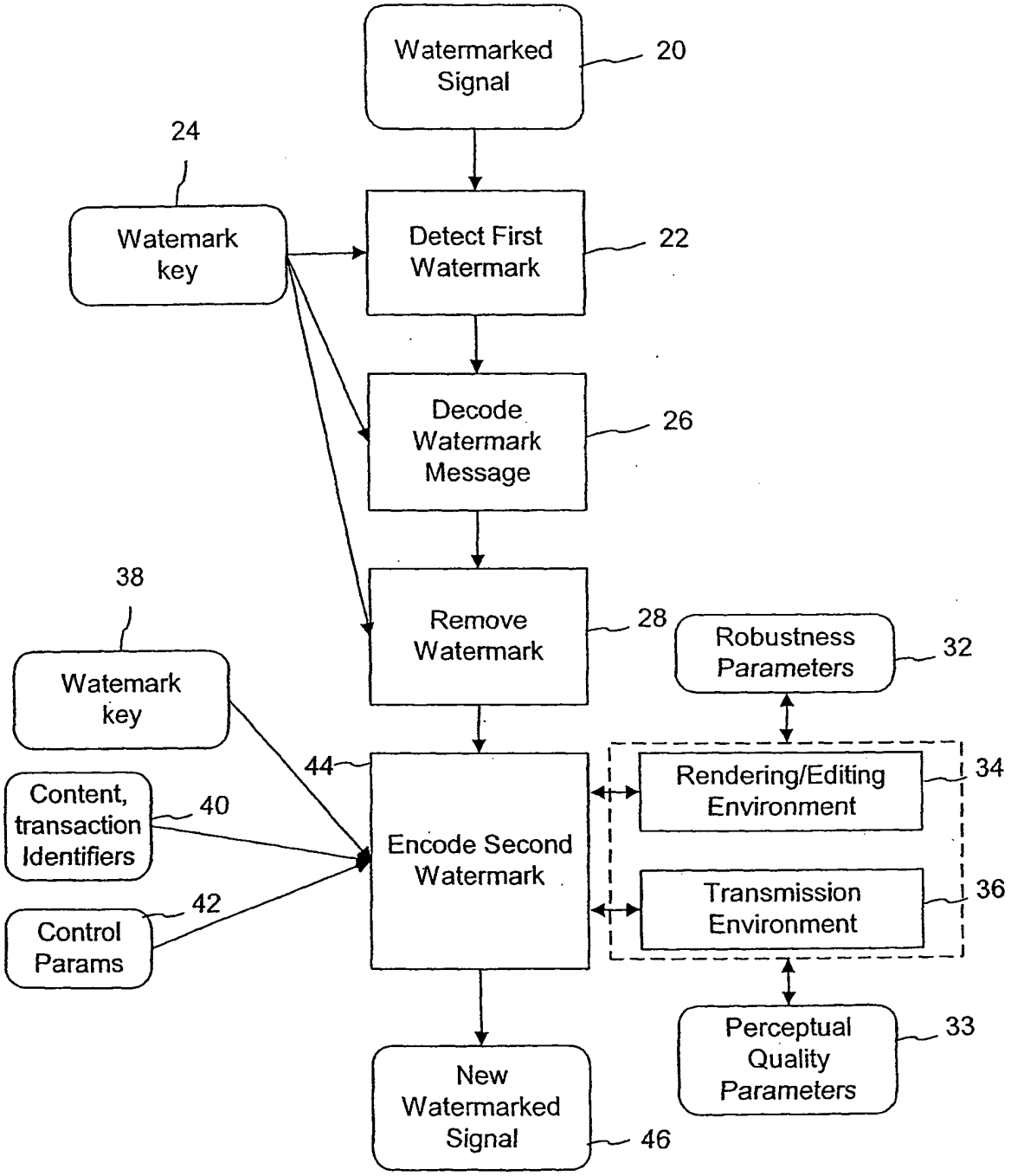

Watermarked Signal
20
24
Watemark key
Detect First Watermark
22
Decode Watermark Message
26
38
Watemark key
Remove Watermark
28
Robustness Parameters
32
44
Content, transaction Identifiers
40
Encode Second Watermark
Rendering/Editing Environment
34
Transmission Environment
36
42
Control Params
Perceptual Quality Parameters
33
New Watermarked Signal
46

Fig. 6

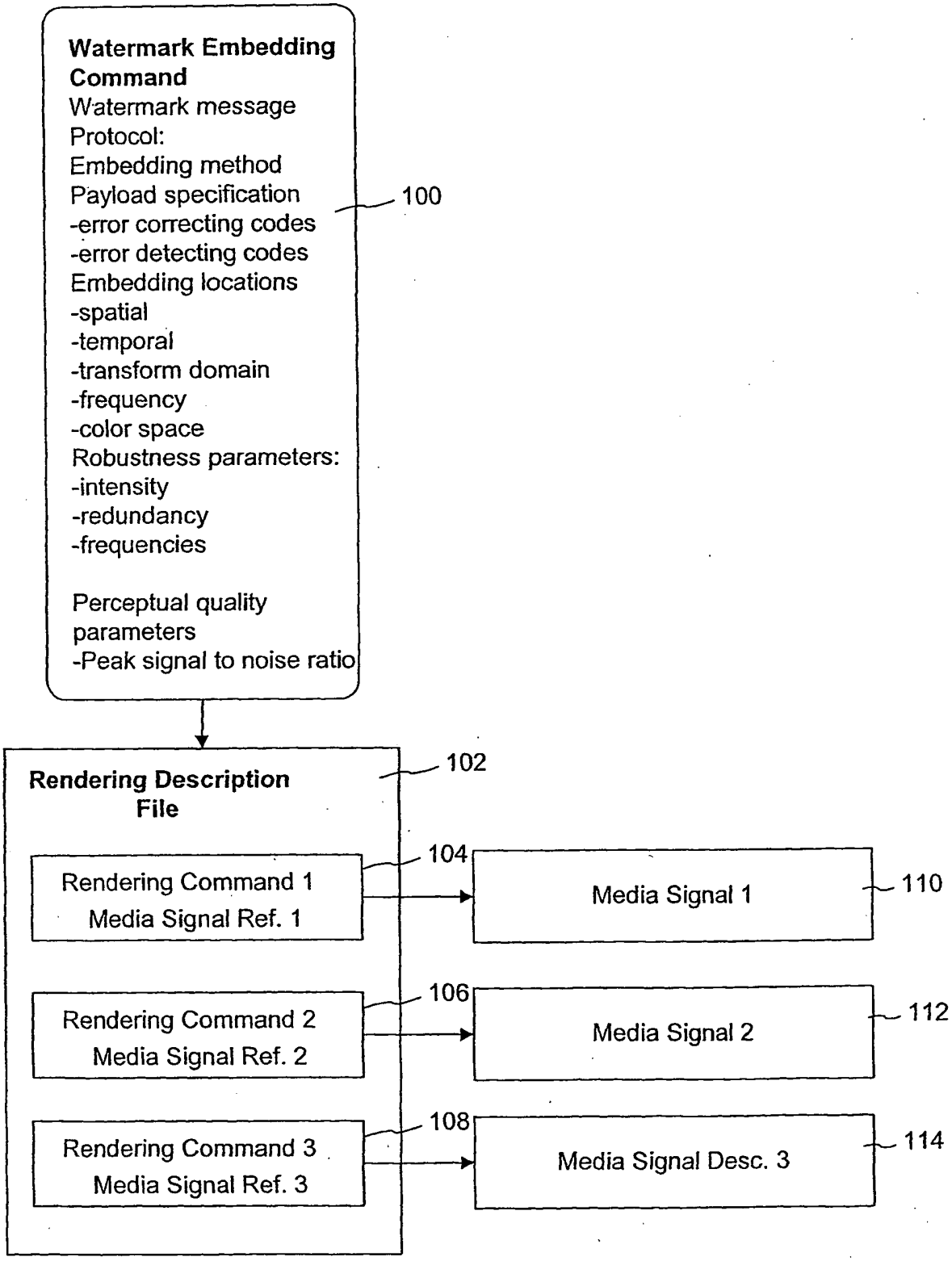
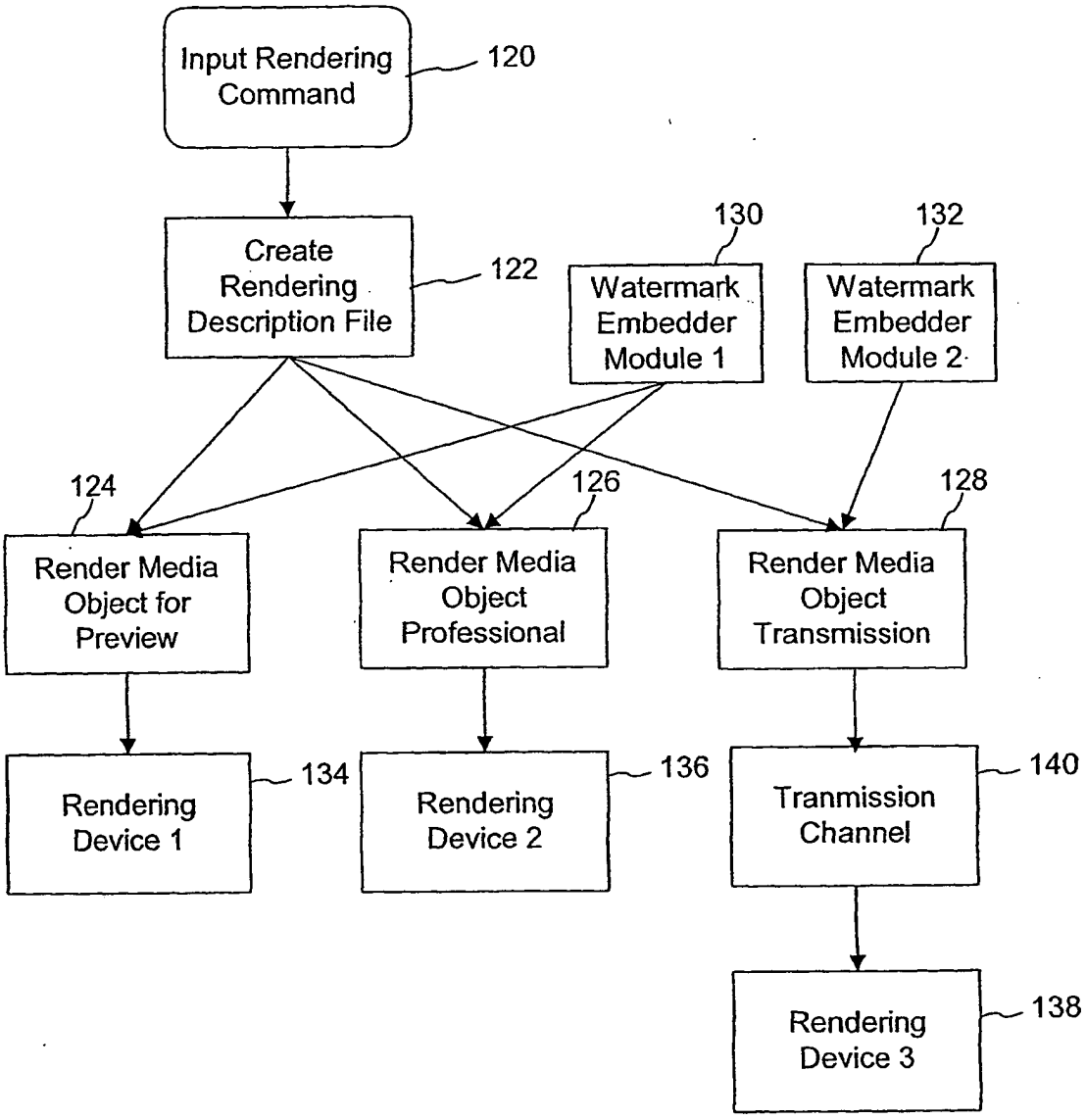

Watermark Embedding Command
Watermark message
Protocol:
Embedding method
Payload specification
-error correcting codes
-error detecting codes
Embedding locations
-spatial
-temporal
-transform domain
-frequency
-color space
Robustness parameters:
-intensity
-redundancy
-frequencies
Perceptual quality parameters
-Peak signal to noise ratio
100
Rendering Description File
102
Rendering Command 1
Media Signal Ref. 1
104
Media Signal 1
110
Rendering Command 2
Media Signal Ref. 2
106
Media Signal 2
112
Rendering Command 3
Media Signal Ref. 3
108
Media Signal Desc. 3
114

Fig. 7


Input Rendering Command
120
Create Rendering Description File
122
130
Watermark Embedder Module 1
132
Watermark Embedder Module 2
124
126
128
Render Media Object for Preview
Render Media Object Professional
Render Media Object Transmission
Rendering Device 1
134
Rendering Device 2
136
Tranmission Channel
140
Rendering Device 3
138

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/08315

A. CLASSIFICATION OF SUBJECT MATTER
IPC(7) : H04K 1/00
US CL : 382/100
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
U.S. : 382/100, 232; 380/210, 252, 287, 54; 713/176; 370/527, 529

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X --- Y	EP 0 0581 317 A2 (INTERACTIVE HOME SYSTEMS) 02 February 1994 (02.02.1994), see the Abstract, page 1, lines 27-49, page 3, line 52 through page 4, line 42.	1-4, 14-18 --- 5-13, 19-20
Y	KIM, W.-G., et al., "A Watermarking Scheme for Both Spatial and Frequency Domain to Extract the Seal Image without the Original Image," Proc. 5th Int. Symp. on Signal Processing and its Applications, August 1999, pp. 293-296, especially the Abstract and Section 3.1, Watermark embedding.	5-11, 19
Y	OHNISHI, J, et al., "A Method of Watermarking with Multiresolution Analysis and Pseudo Noise Sequences," Systems and Computers in Japan, Vol. 29, No. 5, May 1998, pp. 11-19, especially the Abstract and Section 3, Watermarking Algorithm.	11-13, 20
A	MINTZER, F., et al., "If One Watermark is Good, are more Better?," Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, March 1999, pp. 2067-2069.	1-20
A	LU, C.-S., et al., "Highly Robust Image Watermarking Using Complementary Modulations," Proc. 2nd Information Security Workshop, LNCS vol. 1729, November 1999, pp. 136-153.	1-20
A	WO 99/18723 A1 (MACROVISION CORPORATION) 15 April 1999 (15.04.1999), see the entire document.	21-40

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
03 May 2001 (03.05.2001)	03 JUL 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703)305-3230	Authorized officer Andrew W. Johns Telephone No. (703) 305-9900

Form PCT/ISA/210 (second sheet) (July 1998)

INTERNATIONAL SEARCH REPORT

International application No.

PCT/US01/08315

C (Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 0 901 282 A2 (HITACHI, LTD.) 10 March 1999 (10.03.1999), see the entire document	21-40
X	US 5,947,548 A (ADAMS) 26 October 1999 (26.10.1999), Figures 2, 5, 7; see the Abstract, column 4, lines 25-63, column 8, line 32 through column 9, line 28, and column 14, lines 4-65.	41, 51, 58
—		
A		42-49, 50-57, 59-60
A	OHBUCHI, R., et al., "A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces," Proc. Computer Graphics International (CGI'99), June 1999, pp. 180-187.	41-60
A	YI, X, et al., "Agent-Based Copyright Protection Architecture for Online Electronic Publishing," Proc. SPIE vol. 3657: Security and Watermarking of Multimedia Contents, January 1999, pp. 484-493.	41-60